Building Virtual Environments: A 50,000 Foot View

Matthew Conway, Ph.D.

Outline

• Typical Software Architecture for VR
• Modelers
• Texture Painters
How To Build a Virtual World?

- How To Write A Book?
  - Novel?
  - Non-Fiction?
  - Dictionary?
  - Poetry?
  - Pop-up?
  - Field Guide?

Genres of VR

- Entertainment
- Scientific Visualization
- Simulation
- Education
- Remote Manipulation
- Probably others

Different genres require different…

Different genres require different…

Research
Storyboards
Interviews
Prototypes
Artwork/Photos

Artistic Skill
Modeling Talent
Storyboards
Realtime Data Acquisition
Audio Expertise
And Now, A Cartoon.

- This is just a serving suggestion
- Your Virtual Reality May Vary

Software

Scene Graph → Render → Graphics Subsystem → Display devices

Scene Management
Java
Python
C/C++

Low Level Graphics APIs
C/C++

Rendering Hardware

Monitor, Goggles, etc
What Level Of Abstraction?

- Render (Draw)
  - DrawTriangle, SetLineColor, SetLightColor, ReadFrameBuffer, SetShadingModels
- Scene Graph (Database)
  - LoadFile, CreateNode, MoveObject, SetPickable

Scene Graph Example
Scene Graphs Are Much More Than Grouping

- Pushing/Popping Graphics State
  - Materials, Colors, Lights
- Transformations
  - Translations, Rotation, Scale, Shear
- Other
  - Cameras, Sounds, Scripts
So What Does This Get You?

- A Tree Full Of Nodes…
  - Groups
  - Shapes
  - Materials
  - Sounds
  - Scripts
  - Cameras

Traversal = Rendering

- Graph is traversed several times per frame
  - Graphics rendering
  - Sound "rendering"
  - Animation "behavior execution"
  - Input device management
  - Event generation (collision detection)
- The “Draw Frame” command is sometimes implicit (traverse as fast as you can)
Examples

- Java3D
- Inventor
- Performer
- Superscape
- Even Higher Level: WorldUp

Why Is This Good?
Optimizations Are Possible

- High level of abstraction, more power
- Cell-level culling, distance culling, parallelism,
- LOD Control
- Storing, fetching, scheduling of geometry to hardware pipeline(s)
- Implementation is left to graphics experts

Level of Detail

- Provide very low polygon count versions of a model for use when model is at a distance, in fog, moving fast
- Requires a sophisticated runtime to know when to switch models and to prevent “popping”
Software

Startup
- Startup devices
- Load models, textures
- Start the simulation

Simulation loop
1.) Get Tracker Data
2.) Get Events
3.) Update Simulation State
4.) Render State

Trackers
Mic
Buttons
Voice Recognizer
Event Queue

Display(s)
- HMD: 1-2
- CAVE: 4-6

Audio
- Stereo, spatialized?

A Word About Performance

- Every 1/10 second (*30-60 Hz preferred*):
  - Get eye point from tracker
  - Render 3D view (2 views if stereo)
  - Detect Collisions, etc.
  - Call Application Callbacks
- Need low polygon count
- Simple illumination model (Gouraud)
- Optimization (e.g. view culling, LOD)
Simulation

Simulation loop
1.) Get Tracker Data
2.) Get Events
3.) Update Simulation State
4.) Render State

Startup
Startup devices
Load models, textures
Start the simulation

The VR Simulation Frame

- Events
- Clock-Based Animations
- Per-Frame Actions
The VR Simulation Frame

- Events
  - Run Callbacks from the Events In Event Queue

- Clock-based animations
  - Any Value Can Be Animated
  - Position, Orientation, Color,
  - Application variables: e.g. “hunger level for predators”
    - Start=0, End=100, Time=1sec
    - Look at system clock, time elapsed
    - Interpolate value based on time elapsed
    - set values
Linear Interpolation

Value

endval

0.75 x endval

startval

0% 75% 100%

Time as % done

Slow In/Slow Out Interpolation

Value

0.95 x endval

startval

0% 75% 100%

Time as % done
Animation Is Another Talk

- This can be a very complex subsystem
- Interpolation functions
- Pre/post functions
- Predictive Behavior
- Rendering Critical Frames
- Synchronization

Per Frame Actions

- Frame-based animations – often callbacks
  - Collision Detection
  - Numerical simulation step
  - Interobject message passing
Software

Simulation loop
1.) Get Tracker Data
2.) Get Events
3.) Update Simulation State
4.) Render State

Startup
Startup devices
Load models, textures
Start the simulation

Labor

Simulation loop
1.) Get Tracker Data
2.) Get Events
3.) Update Simulation State
4.) Render State

Startup
Load models, textures
Start the simulation

Display(s)
HMD: 1-2
CAVE: 4-6

Audio
Stereo, spatialized?
People: Basic Skill Sets

- Modelers
  - Build 3D objects
- Texture Artists
  - Create Bitmaps, map them to objects
- Audio Designers
  - Compose/mix sound clips
- Programmers
  - Graphics: Draw things fast (Scene Graph)
  - App: Object Behavior, simulation, interface, app logic
  - Sometimes same person, sometimes not

Remember Genres?

- Some Fields Make Less Use of Static Models
  - Scientific Visualization
  - Remote Manipulation
- Some Fields, it is Everything
  - Games
Modeler Goals

- Work Within a low “Polygon Budget”
- Remember Performance Goal of 30 fps
- This is a temporary condition, moving target
- Realtime Raytracing….someday soon?

Polygon Counts

- This guy is approx. 1500 polygons, no texture.
But this guy…19,000+

Modeler Goals (cont)

- Giving objects structure
  - even if no Scene Graph
- Naming parts
  - Names, subparts → communicated to the simulator/behavior engine
  - Modeler work closely with the programmer
  - Texture applied later
Using Object Names In Code

- Later, in the simulation....
  rotate(
  Airplane.left_wing.engine.prop,...
  )
End of Modeling

- Questions?

- Modeling gets you shape. What about Appearance?

Ways to Get Appearance

- Standard Computer Graphics:
  - Color
  - Shading Models
  - Lighting
  - Material
  - Texture
    - Bitmaps wrapped around the shape..
Adding Texture

- Adds high-frequency detail that would be unreasonable to achieve through polygons and color
Textures

- **Artist** creates Bitmaps
- “wrap the bitmap around the model” via UV Mapping
UV Mapping

- Cutting, Stretching, Rotating a flat texture to project on a 3D surface
- Texture Exists in (u,v) space
- Model's polygons carry (u,v) coordinates in addition to (xyz)
- Use tools to assign the proper (u,v) pairs to each polygon
Painting in 3D

- Often labor intensive, but there are tools: DeepPaint, StudioPaint, BodyPaint

Sounds

- Canned Playback
  - Play a WAV file when Event X happens
- Interactive Synthesized Audio
  - Much harder to author
  - Parameterize amplitude, pitch, reverb, anything based on program state
Basic Skill Sets

- Often hard to find one person with more than one or two skills

Summary

- Toolkits are available
- Program low level graphics as a last resort
- Build a team with lots of different talents
- Expect culture clashes
Questions

Yes. fin
A Bunny Model

Hierarchy

- bunny
  - Body
    - Head
      - Left Ear
      - Right Ear
    - Left Arm
    - Right Arm
  - Drum
In 1967…

Disney “Gatorvision”
Other I/O Devices: exotic

- Data gloves
  - Fiber optic bend in fingers

- Pinch Gloves
  - Conductive contact pads on fingers/palm

- Chording keyboards
- Haptic/Force feedback